

Patrones de Usabilidad Temprana en el Modelo Conceptual

Early Usability Patterns in the Conceptual Model

Presentación: 28/08/2020

Doctorando:

Juan Carlos Moreno

Centro de Investigación, Desarrollo y Transferencia de Sistemas de Información (CIDS). Facultad Regional Córdoba.
Universidad Tecnológica Nacional. Argentina
jmoreno33@gmail.com

Director/es:

Marcelo Martín Marciszack

Co-director:

Mario Alberto Groppo

Resumen

Internet ha incrementado el desarrollo y uso de aplicaciones en entorno web. Sin embargo, no todas las aplicaciones web tienen éxito en el mercado. Muchos sistemas son desarrollados con fallas en la usabilidad, que son muy costosas de arreglar una vez que el sistema fue construido. La usabilidad de un sistema de información determina el grado de satisfacción que experimenta un usuario al interactuar con el mismo. Por este motivo, la Usabilidad está fuertemente ligada al ciclo de vida del software y es necesario incorporarla como una característica de calidad a tener en cuenta, durante todo el proceso de desarrollo. Una forma de lograr este objetivo es a través de los patrones de usabilidad. La Ingeniería de Sistemas de Información ha incrementado el empleo de patrones en el modelado conceptual de los sistemas para garantizar no cometer errores y emplear buenas prácticas aprendidas. La siguiente propuesta muestra un conjunto de patrones de usabilidad que pueden incorporarse en el modelado conceptual de los sistemas en forma temprana, para garantizar la presencia de ciertos criterios de Usabilidad desde la etapa de elicitación de los requerimientos.

Palabras clave: Patrón; Modelo de Conceptual; Requerimientos; MDA; Usabilidad.

Abstract

The Internet has increased the development and use of applications in the web environment. However, not all web applications are successful in the market. Many systems are developed with usability flaws, which are very expensive to fix once the system is built. The usability of an information system determines the degree of satisfaction that a user experiences when interacting with it. For this reason, Usability is strongly linked to the software life cycle and it is necessary to incorporate it as a quality characteristic to take into account throughout the development process. One way to achieve this goal is through usability patterns. Information Systems Engineering has increased the use of patterns in the conceptual modeling of systems to ensure that no mistakes are made and use good learned practices. The following proposal shows a set of usability patterns that can be incorporated early into the conceptual modeling of systems, to guarantee the presence of certain usability criteria from the requirements elicitation stage.

Keywords: Pattern; Conceptual Model; Requirements; MDA; Usability.

Introducción

Un aspecto crítico en el desarrollo del software es la definición correcta de lo que se desea construir como sistema de información. La captura de requisitos adquiere un papel relevante, sobre todo cuando se plantean sistemas de alta complejidad donde la especificación de los requisitos es fundamental. De este modo, se observa la importancia de la elicitación de los requerimientos, puesto que permiten obtener un mejor conocimiento del dominio bajo estudio y contribuyen a mejorar el diseño y construcción de un producto, que responda a las necesidades de los usuarios (Qasim, I., Azam, F., Anwar, M. W., Tufail H., Qasim, T., 2018). La reestructuración de un sistema en una etapa tardía puede ser muy costosa y conducir al fracaso del proyecto por llegar tarde al mercado. Para mejorar esta situación, sería beneficioso que el conocimiento relacionado con la usabilidad se capture de una forma que pueda utilizarse para informar el diseño arquitectónico, de modo que las características de usabilidad sean tomadas en cuenta al principio del proceso de diseño. Una herramienta que la comunidad científica desarrolló para mejorar los procesos es la aplicación de patrones. En este contexto se analizará el desarrollo de patrones para la mejora de la usabilidad del software.

Este artículo tiene como objetivo hacer un acercamiento al estado en el que se encuentra la investigación sobre patrones de usabilidad, e identificar cuales patrones de aplicación temprana son de alto impacto en el modelado conceptual de los sistemas. Se trabaja con patrones puesto que contribuyen a no cometer errores por parte del diseñador del software y permite considerar aspectos que pueden haber sido pasado por alto.

El documento se organiza en secciones del siguiente modo: en la sección de Desarrollo se detallan distintos fundamentos teóricos relacionados con un patrón de usabilidad; luego se mencionan los principales patrones de Usabilidad cuyo origen es a partir de distintos trabajos científicos, para finalmente mostrar aquellos que tienen un gran impacto en el diseño del software y son aplicables a distintos escenarios partiendo desde la etapa de requisitos de software, mostrando un ejemplo de una aplicación particular. Finalmente se muestran los resultados obtenidos mediante un ejemplo y se brinda las conclusiones del análisis de la propuesta de trabajo.

Desarrollo

El concepto de patrón fue introducido por Alexander en 1977, en el dominio de la arquitectura. Se buscaba una solución basado en buenas prácticas a un problema considerado recurrente y específico. Cuando el concepto de patrón está referido al dominio de desarrollo del software, su concepción adopta diferentes propósitos. Implementar patrones en el proceso de desarrollo de un sistema es muy útil, porque contribuyen a diseñar mejor los modelos sin errores, estableciendo enlaces más consistentes entre la especificación de requerimientos y el modelado conceptual. El desarrollo de aplicaciones en entornos web ha suscitado la necesidad de incorporar aspectos de calidad relacionados con la usabilidad desde el inicio de la construcción de los sistemas de información. Este objetivo se puede lograr incorporando patrones (de usabilidad) en etapas tempranas del desarrollo de software, mediante la especificación de características inherentes a la Usabilidad en el modelado conceptual de los sistemas (Karttinen, J., Palviainen J. & Koskimies, K., 2007). La norma ISO/IEC 9126-1], considera a la usabilidad como un parámetro de calidad del software y la define como “la capacidad en que un producto de software puede ser entendido, aprendido y usado por determinados usuarios bajo ciertas condiciones en un contexto de uso específico”. La norma que se adopta para analizar este aspecto de calidad (Usabilidad) es la ISO/IEC 25010.

Patrones de Usabilidad

Un patrón de usabilidad se define como una descripción de soluciones, que mejoran los atributos de Usabilidad en un determinado contexto y bajo ciertas condiciones. A partir de este concepto, se analizaron distintos tipos de patrones de Usabilidad, empleados en diversos contextos, momentos del desarrollo y de aplicación de los mismos. Los patrones de Usabilidad que interesaron analizar fueron aquellos que trataban a ciertos criterios usabilidad en etapas tempranas de desarrollo, o aquellos que eran tratados en metodologías que aplican el enfoque de desarrollo conducido por modelos (MDD) con patrones de Usabilidad (Herrera, J.L., Valdenebro, M.D., Juspian, M.A., Arroyo, A.T., & Ordóñez, C.A., 2010). De los trabajos científicos que se analizaron, se obtuvo un listado de patrones que pueden ser especificados y reutilizados en la etapa de elicitación de

requerimientos o en el proceso de diseño del modelado conceptual de un sistema. Dichos patrones brindan soporte y solucionan a distintos problemas específicos relacionados con criterios o subcriterios de usabilidad. Cabe destacar que dichos patrones no buscan solucionar temas relacionados de la interfaz del usuario, sino más bien con el comportamiento del sistema y específicamente de Usabilidad. Esto último permite capturar los mismos como requerimientos desde el punto de vista de la Ingeniería de Sistemas y validar con el usuario los criterios analizados (Ahmed Seffah, Taleb Mohamed, Halima Habieb-Mammar Alain Abran, 2008). A continuación, se detallan cada uno de los patrones de usabilidad evaluados, que se observan en la Tabla 1.

Nombre	Propiedad de Usabilidad que resuelve
Cancel	Error management
Context sensitive help	Guidance
Different Languages	Accessibility
History	Error management
System feed back	Explicit user control
	Provide feedback
	Long action feedback
Undo	Explicit user control
	Error management
User profile	Adaptability
User Input Error Prevention	Structured text entry
Wizard	Step by step solution
Working data visualization	Visual consistency
Reduce risk of errors	Visual consistency
	Operability
Dynamic presentation in user interface	Understandability
	Operability
Progress indicator	User satisfaction
Keep the user focused	Visual consistency
	Understandability
	Operability

Tabla 1: Tabla de Patrones de Usabilidad Analizados (ejemplo de un conjunto) .

Para poder evaluar la presencia de Usabilidad en un producto de software, la misma se descompone en subatributos que son medibles (Holger Röder, 2012). Esto permite vincular los subatributos con aspectos del diseño, que permiten verificar la presencia de la Usabilidad en los modelos conceptuales. El objetivo era vincular los atributos y los subatributos en los que se descompone la Usabilidad con su aplicación en el modelado conceptual mediante el empleo de patrones. Para ello, entonces, se buscó la relación que existe entre los patrones y los subcriterios de Usabilidad que pueden ser empleados en el modelado conceptual de aplicaciones web.

Patrones de Usabilidad de Temprana Relevantes

Es necesario realizar un vínculo entre las propiedades de usabilidad y los criterios. Para ello, se tomó como referencia la norma ISO/IEC 25010, que tiene en cuenta las propiedades de usabilidad desde dos puntos de vista: desde el punto de vista del producto y desde el punto de vista del proceso. Las propiedades de usabilidad se refieren a las características que debe satisfacer un producto de software para ser usable. Una vez identificado dichos atributos, se realizó un vínculo de cada propiedad con distintos patrones, que permitirían resolver problemas relacionados con propiedades específicas de la Usabilidad. Los patrones de Usabilidad proponen soluciones abstractas, que pueden ser incorporadas al modelado conceptual en forma temprana. Los patrones de Usabilidad propuestos representan a conceptos abstractos pero que son aplicables en el diseño. Para poder comprender dicha relación se estructura una tabla, donde se establece el vínculo entre los patrones, las propiedades o atributos que consideran a la usabilidad en forma temprana.

Cabe destacar, sin embargo, que existen patrones que se emplean en etapas tempranas del desarrollo, durante la elicitación de requisitos y que son empleados en forma recurrente en los proyectos. Estos patrones se emplean para capturar un criterio de usabilidad específico. Se expone a continuación cuales son y una breve descripción de los mismos (ver Tabla 2). Dichos patrones, son considerados de alto impacto.

Característica de Usabilidad	Mecanismo de Usabilidad	Objetivo	Alto impacto en la funcionalidad del software	Autores que brindan información sobre esta función
Feedback (Retroalimentación)	System Status	Para informar a los usuarios sobre el estado interno del sistema.	Bass, Juristo.	Tidwell, Welie, Laasko, Brighton, Coram
	Interaction	Para informar a los usuarios que el sistema ha registrado una interacción del usuario, es decir, que el sistema ha escuchado a los usuarios.	(Bass L., John B., Kates. J. , 2001; Juristo N., Moreno A., Sánchez. M., 2008)	
	Warning	Informar a los usuarios de cualquier acción con consecuencias importantes.		
	Long Action Feedback	Para informar a los usuarios que el sistema está procesando una acción que llevará un tiempo completar		
Undo/Cancel (Deshacer/Cancelar)	Global Undo	Para deshacer acciones del sistema en varios niveles	Bass, Juristo.	Brighton, Tidwell, Welie, Laasko
	Object-Specific Undo	Para deshacer varias acciones en un objeto	(Bass L., John B., Kates. J. , 2001; Juristo N., Moreno A., Sánchez. M., 2008)	
	Abort Operation	Para cancelar la ejecución de un comando o una aplicación		
	Go Back to a Safe State	Para volver a un estado particular en una secuencia de ejecución de comando		
Form/Field Validation (Validación de Formulario/Campo)	Structured Text Entry	Para ayudar a evitar que el usuario cometa errores de entrada de datos.	Bass, Juristo. (Bass L., John B., Kates. J. , 2001; Juristo N., Moreno A., Sánchez. M., 2008)	Brighton, Tidwell
Wizard (Asistente)	Step-by-Step Execution	Para ayudar a realizar tareas que requieren diferentes pasos con la entrada del usuario	Bass, Juristo. (Bass L., John B., Kates. J. , 2001; Juristo N., Moreno A., Sánchez. M., 2008)	Welie
User Profile (Perfil de Usuario)	Preferences	Para registrar las opciones de cada usuario para trabajar con el sistema a nivel funcional	Bass, Juristo. (Bass L., John B., Kates. J. , 2001; Juristo N., Moreno A., Sánchez. M., 2008)	Welie, Tidwell
	Personal Object Space	Para registrar las opciones de cada usuario para trabajar con el sistema a nivel de interfaz.		
	Favourites	Para registrar ciertos lugares de interés para el usuario.		
Help (Ayuda)	Multilevel Help	Proporcionar diferentes niveles de ayuda para diferentes usuarios.	Bass, Juristo. (Bass L., John B., Kates. J. , 2001; Juristo N., Moreno A., Sánchez. M., 2008)	Tidwell

Tabla 2: Patrones de Elicitación de Usabilidad

Un patrón de usabilidad se define como una descripción de soluciones, que mejoran los atributos de usabilidad. Las mejoras que ofrecen los patrones presentados a los aspectos de usabilidad, se pueden aplicar posteriormente a las interfaces de los usuarios. Esto da origen a que los mismos se puedan asociar posteriormente con patrones de interfaz o patrones de diseño de interacción. (Garzotto F., Paolini P., Bolchini D., Valenti S., 1999; Folmer, 2003)

Descripción de un Patrón de Usabilidad

A continuación, se describirá como se describirán cada patrón arquitectónico, desde el punto de vista de los escenarios, describiendo la especificación de cada una de las partes que contendrá el mismo. Esto ayudará a describir cada patrón de usabilidad y posibilita la construcción de un catálogo de patrones que resultan útil al diseñador. A su vez, su descripción posibilita la selección en forma inequívoca del patrón, para el problema que se intenta solucionar. La inclusión de patrones de usabilidad en etapas tempranas del diseño del software no es habitual (Ver Gráfico 1). Sin embargo, en los últimos años el desarrollo de aplicaciones web que emplean patrones ha sido incremental, puesto que garantizan no cometer errores e incluyen aspectos de buenas prácticas que pueden no haber relevados en la etapa de elicitación de requerimientos. Se muestra a continuación su estructura:

Definición de Patrón de Usabilidad	Ejemplo de un Patrón de Usabilidad
<ol style="list-style-type: none"> 1. Nombre del Patrón de Usabilidad: El patrón de ser identificable por su nombre y el mismo debe ser lo suficientemente indicativo tanto del problema que abordan como la solución del mismo. 2. Problema: Debe describir cuando aplicar el patrón y en que contexto determinado se debe aplicar. Por ser un patrón de usabilidad, debe referenciar que propiedad de la Usabilidad solucionaría o satisficaría con su aplicación. 3. Solución: La solución describe los elementos que formarán de la solución, las relaciones y vínculos que posean con otros elementos del modelo, las responsabilidades del elemento dentro del modelo. La solución en el modelo conceptual será explicitada de dos maneras distintas: <ol style="list-style-type: none"> 1. Representación gráfica: Una figura representará a un componente o a los compone y las interacciones del mismo, como representación del patrón en la arquitectura. 2. Actores: Describiría cuales son los componentes que participan en la solución propuesta y cuál es su responsabilidad dentro del escenario del modelo propuesto. 4. Ventajas: En esta sección se describirá que aspectos de la usabilidad se se verán beneficiados por la inclusión del patrón en el modelo. 5. Beneficios: En esta sección se incluirán las fortalezas y debilidades que puede proporcionar el patrón por su inclusión en el modelo. Los datos consignados surgirán de la experiencia de su inclusión. 6. Consecuencias: Se describirá el posible impacto que puede tener el patrón en otros atributos de calidad. 7. Patrón/es relacionado/s: Cuales son los patrones que se encuentran relacionados con este patrón y cuáles serán las diferencias de este patrón con los relacionados. 8. Implementación del patrón: Se describirá la implementación del patrón y las clases u objetos que serán derivados y necesarios por la implementación del patrón. 9. Ejemplo: se daría un ejemplo de la aplicación del patrón. 	<ol style="list-style-type: none"> 1. Nombre del patrón: Retroalimentación 2. Problema: proporcionar información relacionada con el estado del sistema. 3. Solución: <ol style="list-style-type: none"> 3.1. Representación gráfica: La representación del modelo depende de la metodología a emplear 3.2. Actores: <p>Procesos activos i: Existen varios procesos que están en ejecución y solicitan retroalimentación, informándole a Retroalimentación la información que requieren que les sea retroalimentada.</p> <p>Retroalimentación: El artefacto o módulo es el encargado de recibir las peticiones y del envío de datos. Las flechas indicarán el tipo de retroalimentación solicitado y los datos que se deben retroalimentar desde cada proceso activo. Describe quien será el destinatario de la información retroalimentada, que será enviada a otra parte del sistema de información (interfaz) para informar al usuario.</p> <p>Interfaz: La interfaz será quien reciba la información y muestre la misma al usuario.</p> <p>Sistema: Representa a otras partes del sistema que requieren de información de retroalimentación.</p> 4. Ventajas: A través de este patrón, el sistema proporciona información sobre el estado del sistema y sobre las operaciones que está llevando a cabo, como retroalimentación al usuario. 5. Beneficios: Este patrón contribuye con el atributo de satisfacción de la usabilidad, puesto que la retroalimentación le proporciona al usuario información sobre los procesos del sistema. 6. Consecuencias: Este patrón impide que se ejecuten procesos que sobrecarguen el sistema, evitando que se ejecuten procesos que vuelvan inestable el sistema. Ayuda a mejorar la mantenibilidad del sistema, puesto que la información está canalizada. 7. Patrones relacionados: Implementación: Este patrón se implementa como un patrón dedicado a informar al usuario y a los sistemas sobre lo que está sucediendo con respecto a los procesos en ejecución. Los sistemas notifican al patrón Retroalimentación sobre lo que está sucediendo en el sistema. 8. Implementación del patrón: Se describirá la implementación del patrón y las clases u objetos que serán derivados y necesarios por la implementación del patrón. 9. Ejemplo: se daría un ejemplo de la aplicación del patrón.

Gráfico 1: Ejemplo de la Definición de un Patrón de Usabilidad Temprana.

Resultados

Una vez comprendidos los patrones que pueden emplearse en el modelado conceptual de los sistemas, se elaboró un catálogo de patrones, que brindan al analista de sistemas de información dos aspectos como ventajas: a) la facilidad de poder seleccionar un patrón de usabilidad, que está en función de un atributo del mismo que se desea mejorar en el modelo conceptual para un uso específico; b) asimismo, además, brinda información sobre

el impacto que puede provocar sobre otros atributos la inclusión de dicho patrón en el modelo. Estos aspectos representan una contribución importante para quien diseña sistemas.

Conclusión

Los patrones de usabilidad son empleados en el modelado conceptual como una solución para mejorar ciertos aspectos de calidad, que se carecen o necesitan ser mejorados en el modelado conceptual de los sistemas. La generación de patrones de usabilidad conlleva a plantear la necesidad de establecer una guía de buenas prácticas para la selección de patrones desde un catálogo, el cual debe reunir ciertas características tales como: a) poder seleccionar un patrón de usabilidad de forma precisa, ya sea por los atributos que el patrón intenta solucionar y qué impacto puede producir su introducción en el modelo; b) como emplear el patrón de usabilidad seleccionado, en un modelo conceptual de un dominio determinado.

La incorporación de forma temprana, implica que los requerimientos de Usabilidad deben ser capturados desde la etapa de elicitación de requerimientos. Esto trae aparejado la inclusión de otras propiedades de usabilidad debido a que muchas de estas propiedades están vinculadas unas con otras. La idea de considerar la usabilidad desde la etapa de elicitación de requerimientos influirá en el desarrollo y diseño de la arquitectura del software, y en su posterior implementación tras la adopción de un lenguaje de programación. Esto contribuirá a disminuir los tiempos de reprocesos, disminuirá los costos y ayudará a obtener un software de calidad.

Referencias

- Ahmed Seffah, Taleb Mohamed, Halima Habieb-Mammar, Alain Abran, "Reconciling usability and interactive system architecture using patterns", *Journal of Systems and Software* Volume 81, Issue 11, Pages 1845-1852, 2008
- Bass L., John B., Kates. J. Achieving Usability Through Software Architecture. Technical Report. CMU/SEI-2001-TR-005, March 2001.
- Brighton, P. Usability Pattern Collection. [Http://www.cmis.brighton.ac.uk/research/patterns](http://www.cmis.brighton.ac.uk/research/patterns)
- Folmer, E. (2003). Usability Patterns in Software Architecture.
- Garzotto F., Paolini P., Bolchini D., Valenti S. (1999) "Modeling-by-Patterns" of Web Applications. In: Chen P.P., Embley D.W., Kouloumdjian J., Liddle S.W., Roddick J.F. (eds) *Advances in Conceptual Modeling*. ER 1999. Lecture Notes in Computer Science, vol 1727. Springer, Berlin, Heidelberg
- Holger Röder, "Specifying usability features with patterns and templates", *First International Workshop on Usability and Accessibility Focused Requirements Engineering (UsARE)*, 2012
- Herrera, J.L., Valdenebro, M.D., Juspian, M.A., Arroyo, A.T., & Ordóñez, C.A. (2010). Architectural patterns regarding web application domain usability
- Juristo N., Moreno A., Sánchez. M. Techniques and Patterns for Architecture-Level Usability Improvements. Deliverable 3.4. STATUS. project. [Http://www.ls.fi.upm.es/status](http://www.ls.fi.upm.es/status) May 2003.
- Kaartinen J., Palviainen, J, Koskimies, K., "A Pattern-Driven Process Model for Quality-Centered Software Architecture Design – A Case Study on Usability-Centered Design", *Australian Software Engineering Conference (ASWEC'07)*, 2007
- Qasim, I., Azam, F., Anwar, M. W., Tufail H., Qasim, T. "Mobile User Interface Development Techniques: A Systematic Literature Review," 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, 2018, pp. 1029-1034, doi: 10.1109/IEMCON.2018.8614764.
- Laasko, S. A.. *User Interface Designing Patterns*, 2003. [http://www.cs.helsinki.fi/u/salaakso/patterns/index tree.html](http://www.cs.helsinki.fi/u/salaakso/patterns/index%20tree.html)
- Tidwell, J. *The Case for HCI Design Patterns*. [Http://www.mit.edu/jdidwell/common ground onefile.htm](http://www.mit.edu/jdidwell/common%20ground%20onefile.htm)
- Tidwell, J. *UI Patterns and Techniques*. [Http://time-tripper.com/uipatterns](http://time-tripper.com/uipatterns)
- Welie, M. Van. *The Amsterdam Collection of Patterns in User Interface Design*. [Http://www.welie.com](http://www.welie.com) .